



Full length article

Generating evolving region embedding with memory-based graph for dynamic urban sensing

Yi Xu ^{a,b}, Zerong Deng ^b, Tongyu Zhu ^{a,b}, Liangzhe Han ^{b,*}, Leilei Sun ^{a,b}, Zhuo Chen ^c, Hao Sheng ^{a,b}

^a Data Science and Intelligent Computing Laboratory, Hangzhou International Innovation Institute, Beihang University, Hangzhou, Zhejiang 311115, PR China

^b State Key Laboratory of Complex & Critical Software Environment(CCSE), Beihang University, Beijing, China

^c China Mobile Information Technology Center, Beijing, China

ARTICLE INFO

Keywords:

Urban region embedding
Dynamic graph
Human mobility
Urban sensing

ABSTRACT

Recently, learning region embeddings from mobility data has become a prevalent solution to various smart city tasks. However, existing methods primarily focus on learning static embeddings, which often fail to capture temporal dynamics critical for tasks like travel time estimation and dynamic crime prediction. To fill the gap, this paper introduces an Evolving Urban Region Embedding method (**EvolveURE**), a generic and evolving region embedding framework tailored for dynamic urban sensing tasks. **EvolveURE** introduces a memory-based embedding module that continuously updates region embeddings by encoding time-aware region-region interactions and recursively updating region memories. Additionally, a time-aware mobility graph reconstruction task is designed to capture detailed temporal information by reconstructing both the magnitude and temporal density of mobility data. To enhance embedding stability and generalization, a cross-scale embedding reconstruction task is proposed, leveraging insights from different data scales and parameter updating mechanisms. The learned region embeddings are tested on diverse downstream urban sensing tasks. Experimental results demonstrate the proposed approach achieves superior results on dynamic urban sensing tasks thanks to these time-aware designs. Codes have been released on <https://github.com/rabbity1999/EvolveURE/>.

1. Introduction

Urban representation learning from human mobility [1] data contains rich semantics about city regions. Such representations can support a variety of urban sensing tasks [2], such as crime prediction [3], flow prediction [4], and travel time estimation [5], which will further contribute to city management [6].

Existing works mainly learn general urban region embeddings from single or multiple geographic graphs (see Fig. 1). HDGE [7] uses human mobility to construct the flow transition graph. ZE-MOB [8] considers the co-occurrence of origins and destinations in human mobility data. Other models like MVURE [9], MVGCL [10] and HREP [11] propose to learn from cross-domain data, where a multi-source data fusion approach is proposed followed by multi-view graph learning to obtain region representations.

The above urban region embedding methods generate static region representations to solve urban sensing tasks with low dynamic requirements, like predicting the total number of crimes per year or land

usage. However, some properties of regions may change over time and many urban sensing tasks have strong temporal dynamics. For example, the crime rates of regions are different from day to day. Compared to predicting yearly, it may be more valuable for city management to predict the regional daily crime number.

To achieve one specific type of urban sensing task with temporal dynamics, taking traffic flow prediction as an example, some methods [12–15] began to organize human mobility data as sequences of snapshots and use sequential learning methods like RNNs [16], CNNs [17], to learn temporal dynamics between these snapshots. Each snapshot is a static graph constructed with mobility data in a period, which reflects the aggregated graph structure information within a fixed time window. However, the discrete-time division is unavoidable to ignore fine-grained mobility information, such as the detailed timestamp.

To tackle the above issues, we propose the **Evolving Urban Region Embeddings (EvolveURE)** model for urban sensing tasks that have a

* Corresponding author.

E-mail addresses: xuyee@buaa.edu.cn (Y. Xu), dengzerong@buaa.edu.cn (Z. Deng), zhutongyu@buaa.edu.cn (T. Zhu), liangzhehan@buaa.edu.cn (L. Han), leileisun@buaa.edu.cn (L. Sun), chenzhuoit@163.com (Z. Chen), shenghao@buaa.edu.cn (H. Sheng).

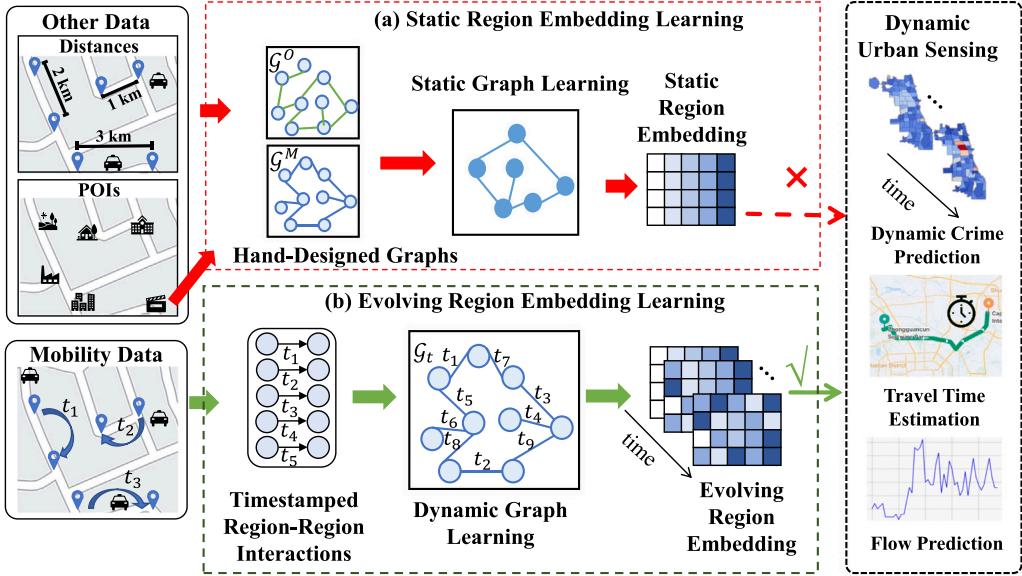


Fig. 1. Existing urban region embedding methods mainly generate static region representations, while we propose to learn evolving region embeddings for dynamic urban sensing tasks.

higher requirement of capturing dynamics, like dynamic crime prediction and travel time estimation. We propose to learn directly from the original mobility data to avoid the information loss incurred by the construction of static graphs. The region-region interactions reflect the latent spatial relation, through encoding interactions continuously, the region-wise spatial relations can be learned dynamically. An evolving region embedding module is designed to encode the chronological region-region interactions into time-aware messages and further use these messages to update the maintained region memories. Based on the continuously updated region memories, the evolving region embedding module will generate evolving region embeddings for final dynamic urban sensing tasks. To ensure the evolving embeddings learn detailed temporal information, a decoder is proposed to reconstruct the time-aware mobility graph with the generated region embeddings. Meanwhile, to reduce the data noise and make the learning process stable, a cross-scale embedding learning module is designed to learn the similarity between region embeddings from different temporal scales.

The contributions of this work are summarized as follows:

- **A generic and evolving region embedding framework** is proposed to support a variety of dynamic urban sensing tasks. Different from previous works that learn static embedding from multiple pre-defined graphs for static urban sensing tasks, this work focuses on generating evolving region embeddings, which benefits a broader range of downstream tasks.
- **A memory-based region embedding module** is designed to generate evolving region embeddings with the newly occurred region-region interactions. The region-region interactions are encoded into time-aware messages. A memory-based updating method is devised to fuse the message with previous region memories, recursively updating the region memories.
- **A time-aware mobility graph reconstruction task** is designed to reconstruct input in feature space. It reconstructs not only the amount but also the temporal density of mobility data.
- **A cross-scale embedding reconstruction task** is devised to reconstruct input in embedding space. Learning from two different data scales and parameter updating mechanisms enhances the stability of learned region embeddings and facilitates a more profound understanding of the essential characteristics of regions.

2. Related works

2.1. Graph representation learning

The aim of graph representation learning is to obtain node representations from non-Euclidean graph structures [18,19]. Early works mainly learn graph representations from a static view [20,21]. Node2Vec [22] transforms the irregular graph as random walks and learns node embeddings from these sequences. GCN [23] transfers the convolution operation from images to graphs. GraphSAGE [24] introduces inductive setting by learning the neighborhood structure of nodes. However, most works assume that the properties of nodes and graph structure will not change over time. To address this, researchers begin to divide the historical data into snapshots [25]. Each snapshot is a static graph, which reflects the aggregated graph structure information within a fixed time window. CNNs [12,13], RNNs [14,15], or attention mechanisms [26,27] are utilized to capture the temporal evolutionary patterns between snapshots. However, the discrete-time window division is unavoidable to ignore fine-grained edge information, such as the detailed timestamp of each edge. To solve this, researchers refined the data processing granularity to per interaction and proposed continuous-time dynamic graph representation learning [28–32]. Although existing works have achieved success in academic networks and social networks, graph representation learning to generate dynamic and ubiquitous embeddings of urban regions faces the challenge of denser interactions and uncertain downstream tasks.

2.2. Region representation learning

With the emergence of smart city development, urban region representation learning from human mobility has received extensive attention from researchers [33–41].

Region representation learning uses low-dimensional vectors to represent the rich properties of urban regions, such as region functions, traffic flows, etc. HDGE [7] uses human mobility to construct the flow transition matrix to learn region representations. ZE-MOB [8] considers the co-occurrence of origins and destinations. MVURE [9] argues that the above methods only use single human mobility data, and the effectiveness of region representations is limited. Therefore, a multi-source data fusion approach is proposed followed by multi-view

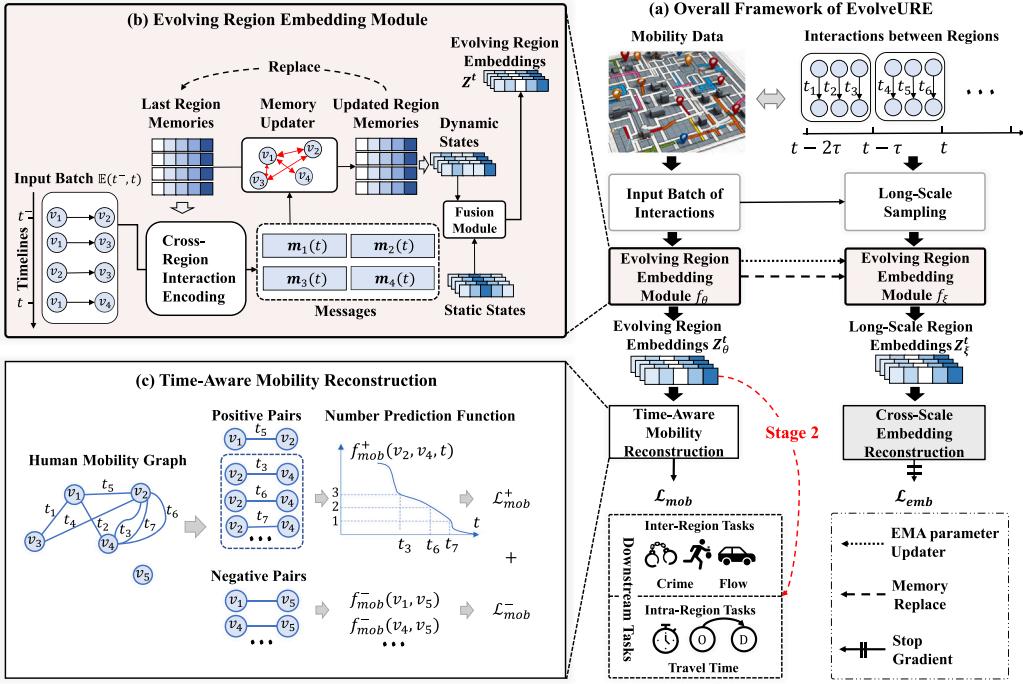


Fig. 2. The pipeline of EvolveURE.

graph learning to obtain region representations. MVGCL [10] introduces graph contrastive learning to enhance the consistency of region representations. HREP [11] combines region representation learning with prompt learning to make representations better adapted to different downstream tasks. These urban region embedding models all generate static region representations to solve urban sensing tasks with low dynamic requirements, such as the prediction of total crime numbers in a year. However, some properties of regions change over time and many urban sensing tasks have strong temporal dynamics such as regional flow prediction, e.g., residential region has high flow in the morning and evening and low flow in the middle of the day. It is also more meaningful for urban management to predict regional crime rates at the granularity of days.

Static region representations fail to reflect varying region attributes. Therefore, it is a valuable question of how to model evolving embeddings to represent time-varying region attributes.

3. Preliminaries

Definition (Human Mobility Graph). Human mobility data can be organized into a human mobility graph, where nodes represent regions within the city, and edges denote the trips between regions. The Human Mobility Graph is a graphical representation of human mobility behavior data, primarily comprising trips across regions in the city. We use $\mathcal{G}^t = (\mathbb{V}, \{e_m | t_m < t\})$ to represent the human mobility graph before time t . \mathbb{V} denotes the node set. $\{e_m | t_m < t\}$ denotes the edge set that consists of trips occurred before t . Each trip $e_m = (v_i, v_j, t_m)$ is characterized by its starting point v_i , destination point v_j , and timestamp t_m .

3.1. Urban region embedding

Considering the human mobility graph up to time t , represented as $\mathcal{G}^t = (\mathbb{V}, \{e_m | t_m < t\})$, our objective is to train an encoder f such that

$$f(\mathcal{G}^t) = Z^t = z_1^t, z_2^t, \dots, z_N^t, \quad z_i^t \in \mathbb{R}^d, \quad (1)$$

where $z_i^t \in \mathbb{R}^d$ represents the embedding of region i at time t .

3.2. Downstream urban sensing tasks

Given the generated urban region embedding Z^t , there are two types of downstream tasks are considered: intra-region and inter-region downstream tasks. Intra-region tasks predict attributes of a single region, like the crime/traffic flow of a certain region: $X_i^{T,t:t+\tau} = f^T(z_i^t)$, T represents different tasks, τ is the time interval. Inter-region tasks involve two or more regions, such as the travel time estimation: $X_{i,j}^{T,t:t+\tau} = f^T(z_i^t, z_j^t)$. In this paper, the learned urban region embeddings are tested on both intra-region and inter-region tasks.

4. Methodology

The framework of EvolveURE is drawn in Fig. 2. It consists of three main parts in stage 1 (generating region embeddings). We will introduce these three parts step by step.

4.1. Evolving region embedding module

Different from methods that learn spatial patterns from multiple static pre-constructed mobility graphs or sequences of snapshots, we directly learn region-wise spatial relations from the region-region interactions that contain the latent spatial relation, extracting information inherently in the raw mobility data. Considering the dense interactions and frequent updating of region states, a memory-based region embedding module is designed, shown in Fig. 3.

Region Memories. We use memory to store the current states $s_i^t \in \mathbb{R}^d$ of each region v_i . To encode the human mobility data, the region memory will be updated by sequences of trip records chronologically. Specifically, the updating of region memories consists of two steps: cross-region interaction encoding and region memory updating. Finally, the evolving region embeddings will be generated with continuously updated region memories.

Cross-Region Interaction Encoding. The original human mobility data are formed into a chronological sequence of interactions between different regions. The region-region interactions reflect the latent spatial relation. Through encoding interactions continuously, the region-wise spatial relations can be learned in a dynamic manner. Once a trip

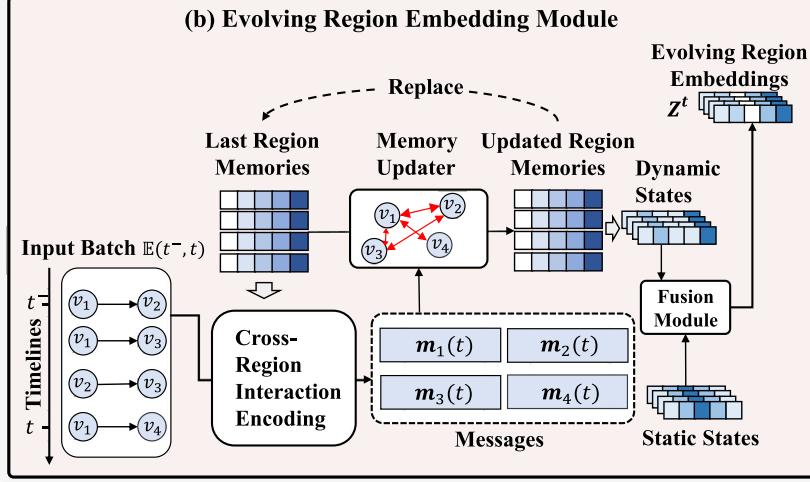


Fig. 3. Illustration of the evolving region embedding module.

from region v_i to v_j happens at time t , this trip can be seen as an interaction between the two regions. The interaction can be encoded into message $m_i(t)$:

$$m_i(t) = (s_i(t^-) \| s_j(t^-) \| \Delta t \| f_j^{node}), \quad (2)$$

where $s_i(t^-)$ and $s_j(t^-)$ are old states of the region v_i and v_j , Δt is the time difference between the interacting time and the last memory updated time of region v_i , f_j^{node} is the node feature of the region v_j . Considering computing efficiency, all messages generated by interactions in current input batch $\mathbb{E}(t^-, t)$ will be aggregated into a region-wise message in a time-weighted manner:

$$m'_i(t) = \sigma(\mathbf{W}_m \frac{\sum_{(v_i, v_j, t) \in \mathbb{E}(t^-, t)} e^{-(t-t^-)} m_i(t)}{\sum_{(v_i, v_j, t) \in \mathbb{E}(t^-, t)} e^{-(t-t^-)}} + b_m), \quad (3)$$

where $e^{-(t-t^-)}$ denotes the time decay from interaction happening time to the last updated time of region v_i . $m'_i(t)$ represents the aggregated messages for region v_i at time t .

Memory Updater. The generated region-wise messages $m'_i(t)$ will be used to update the corresponding region memory recurrently:

$$\begin{aligned} g_i &= \sigma(\mathbf{W}_i \cdot [s_i(t^-), m'_i(t)] + b_i), \\ g_f &= \sigma(\mathbf{W}_f \cdot [s_i(t^-), m'_i(t)] + b_f), \\ s'_i(t) &= \tanh(\mathbf{W}_{h_m} m'_i(t) + g_i \odot (\mathbf{W}_{h_s} s_i(t^-) + b_{h_s}) + b_s), \\ s_i(t) &= (1 - g_f) \odot s_i(t^-) + g_f \odot s'_i(t). \end{aligned} \quad (4)$$

After the updating, the memories of one node will be modified by its historically interacted nodes. And the more frequent and recent the interaction is, the more influence on the central node is. Thus, this procedure can encode the human mobility data into evolving node memories.

Fusing Module. To generate the final region embeddings Z_i^t , static embeddings h_i^{static} are combined with the updated memories $s_i(t)$ which contains the evolving region status of region v_i :

$$Z_i^t = \sigma(\mathbf{W}_z (\lambda h_i^{static} + (1 - \lambda) s_i(t))), \quad (5)$$

where λ is a learnable parameter as the weight controller. The static embeddings h^{static} are represented as a learnable matrix. It is initialized randomly and refined throughout the training process.

Through the evolving region embedding module, the temporal information in region embeddings can be derived from fine-grained interactions, and the spatial relationships are self-learned from tens of thousands of interactions between regions, eliminating the need for manually constructing static mobility graphs.

4.2. Time-aware mobility reconstruction

In the previous subsection, historical mobility data are encoded into evolving region embeddings. Another problem is how to train the model with a useful training objective. Here, a pretext task is designed to reconstruct the mobility graph as detailed as possible using the region embeddings. During this procedure, the finer the granularity of reconstructed mobility data, the richer the semantics that the region embedding has learned from the raw data. In contrast to prior studies [9,11,42] that primarily focused on statistical aspects of data reconstruction, our approach aims not only to restore the quantity of mobility but also to reconstruct the occurring timestamp of interactions.

For any pair of regions (v_i, v_j) , if they have interactions in the current input batch $\mathbb{E}(t^-, t)$, the interactions can be considered as positive pairs. We aim to accurately predict the time and quantity of edges occurring between two regions. If they do not have an edge during this period, they are considered a negative pair. In this case, we just aim to predict the number of edges as close to 0 as possible.

Based on this idea, we can design functions for positive pairs and negative pairs separately. For positive pair $e_m = (v_i, v_j, t_m)$, we use a function to predict the number of trips from region v_i to v_j during the time window $[t_m, t]$:

$$f^+(v_i, v_j, t_m) = \sigma(\mathbf{W}_r^+ [Z_i^t, Z_j^t, t - t_m] + b_r^+). \quad (6)$$

The mobility reconstruction loss for positive pairs is:

$$\mathcal{L}_{mob}^+ = \sum_{e_m \in \mathbb{E}(t^-, t)} \frac{(Y_{ij}(t_m, t) - f^+(v_i, v_j, t_m))^2}{|Y_{ij}(t^-, t)|}, \quad (7)$$

where t^- is the last updated time, $\mathbb{E}(t^-, t)$ is current input batch, and $Y_{ij}(t_m, t)$ is denoted as the true number of trips from region v_i to v_j during time window $[t_m, t]$. Taken mobility graph shown in Fig. 4 as an example, there are 7 positive pairs. If the above task is well solved between v_2 and v_4 , the accurate time and the amount of trips can be also reconstructed by the cumulative curve.

For negative region pair (v_p, v_q) , the function to predict the number between two regions is calculated as:

$$f^-(v_p, v_q) = \sigma(\mathbf{W}_r^- [Z_p^t, Z_q^t] + b_r^-). \quad (8)$$

The mobility reconstruction loss for negative pairs is:

$$\mathcal{L}_{mob}^- = \sum_{(v_p, v_q) \in \mathbb{E}^0(t^-, t)} (f^-(v_p, v_q) - 0)^2, \quad (9)$$

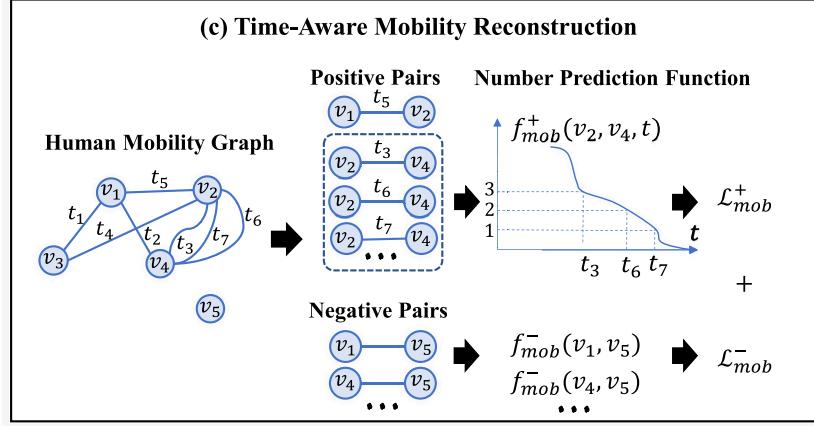


Fig. 4. Illustration of time-aware mobility reconstruction module.

where $\mathbb{E}^0(t^-, t)$ is the set of region pairs that do not have interactions during time window $[t^-, t]$.

Then, the final time-aware mobility reconstruction loss can be defined as:

$$\mathcal{L}_{mob} = \frac{1}{|\mathbb{E}(t^-, t)|} \mathcal{L}_{mob}^+ + \frac{1}{|\mathbb{E}^0(t^-, t)|} \mathcal{L}_{mob}^- \quad (10)$$

4.3. Cross-scale embedding reconstruction

In the preceding section, the time-aware mobility reconstruction module calculates the reconstruction loss in feature space. However, due to noise present in some data, the learned embeddings exhibit perturbations. Simultaneously, the input data segmentation of different lengths and granularities introduces significant fluctuations in the region representations. The learned embeddings are influenced heavily by local temporal patterns as well. To address the problem brought by data noise in the feature space, we further introduce a reconstruction task in the latent embedding space.

The basic idea of the embedding reconstruction task is that the embeddings generated by augmented input data (containing data noise) and the original input data are supposed to be similar. Here, we design a temporal data augmentation to transform input data into a longer time scale. Then, another pretext task is designed to make representations learned from data on different time scales as similar as possible.

To achieve this, we add a parallel branch to encode the augmented input data. First, the input data $\mathbb{E}(t^-, t)$ for the current batch are expanded to a longer temporal scale.

$$\hat{\mathbb{E}}(t^-, t) = \text{Temporal Expansion}(\mathbb{E}(t^-, t), \alpha), \quad (11)$$

where α is the temporal scaling factor. For example, if $\alpha = 1$ and $\tau = t - t^-$, then $\hat{\mathbb{E}}(t^-, t)$ is $\mathbb{E}(t^- - \alpha\tau, t + \alpha\tau) = \mathbb{E}(t^- - \tau, t + \tau)$. The augmented data for long-scale are randomly sampled from this:

$$\mathbb{E}_\xi(t^-, t) = \text{EdgeSampling}(\hat{\mathbb{E}}(t^-, t), 1 + \beta). \quad (12)$$

Ultimately, this yields a long-term input $\mathbb{E}_\xi(t^-, t)$ with a quantity of $1 + \beta$ times the current input edges $\mathbb{E}(t^-, t)$. Long-scale sampling serves as a data augmentation technique designed to enrich the training data while maintaining its inherent diversity. This method focuses on capturing extended temporal dependencies, ensuring that the augmented data reflects the same statistical properties as the original data. By achieving this balance, long-scale sampling not only enhances the dataset but also avoids introducing errors or biases associated with artificially constructed data. Then, the scaled input data $\mathbb{E}_\xi(t^-, t)$ represents the long-term input data, which will be encoded into embedding Z_ξ^t by a long-term region embedding module f_ξ . f_ξ has the same structure with

the original region embedding module f_θ , which takes the original data $\mathbb{E}(t^-, t)$ as input.

Differently, f_ξ is updated by an Exponential Moving Average (EMA) [43] with decay rate γ :

$$\xi \leftarrow \gamma\xi + (1 - \gamma)\theta, \quad (13)$$

where θ represents the parameters in the evolving region module f_θ and ξ represents the parameters in the long-term region module f_ξ . And f_θ is updated through gradient descent:

$$\theta \leftarrow \text{optimize}(\theta, lr, \mathcal{L}). \quad (14)$$

The optimization objective of the cross-scale learning is:

$$\mathcal{L}_{emb} = 1 - \frac{Z_\xi^t(p_\theta(Z_\theta^t)^T)}{\|Z_\xi^t\| \|p_\theta(Z_\theta^t)\|}, \quad (15)$$

where p_θ is a predictor to predict Z_ξ^t with Z_θ^t .

Through the design of such a cross-scale task, representations learned from data across different temporal scales can be made as similar as possible. This may help mitigate the interference of local data noise on the learned representations, which will further promote the stability of learned region representations.

4.4. Model training

The final objective function in stage 1 consists of two parts:

$$\mathcal{L} = \mathcal{L}_{mob} + \mathcal{L}_{emb}, \quad (16)$$

where the first part represents the reconstruction in feature space and the second part can be seen as the reconstruction in the embedding space.

For a better understanding of the training process, we have added the pipeline of stage 1. Shown in Algorithm 1 is the pipeline of stage 1. For stage 1, the time complexity of our proposed method is $O(N_{epoch} |\mathbb{E}^{train}|)$. Specifically, N_{epoch} is the number of total training epochs in stage 1. $|\mathbb{E}^{train}|$ is the number of original region-region interactions in training data.

5. Experiments

In this section, we test the learned evolving region embeddings on multiple urban sensing tasks. Additionally, we mainly want to answer these questions with experiments:

- **RQ1:** How does EvolveURE compare to different state-of-the-art approaches?

Algorithm 1 : Stage 1 of EvolveURE

Input: newest batch of region-region transactions, $\mathbb{E}(t^-, t)$.
Output: region embeddings Z'

```

1: for each epoch do
2:    $\mathbb{E}_\xi(t^-, t) \leftarrow$  Long-Scale Sampling( $\mathbb{E}(t^-, t)$ )
3:    $s_\theta(t), Z'_\theta \leftarrow$  Evolving Region Embedding Module  $f_\theta(s_\theta(t^-), \mathbb{E}(t^-, t))$ 
4:    $s_\xi(t), Z'_\xi \leftarrow$  Evolving Region Embedding Module  $f_\xi(s_\xi(t^-), \mathbb{E}_\xi(t^-, t))$ 
5:   Calculate  $\mathcal{L}_{mob}$  by Time-Aware Mobility Reconstruction( $\mathbb{E}(t^-, t), Z'_\theta$ )
6:   Calculate loss  $\mathcal{L}_{emb}$  by Eq. (15)
7:   Final loss  $\mathcal{L} = \mathcal{L}_{mob} + \mathcal{L}_{emb}$ 
8:   Update parameters of  $f_\theta$  by gradient Descent
9:   Update parameters of  $f_\xi$  by EMA mechanism
10:  Replace the stored memory of  $f_\xi$  with  $s_\theta(t)$ 
11: end for
12: return region embeddings  $Z'$ 
13: procedure EVOLVING REGION EMBEDDING MODULE( $s(t^-), \mathbb{E}(t^-, t)$ )
14:   for each region-region interaction  $(v_i, v_j, t) \in \mathbb{E}(t^-, t)$  do
15:     Get messages  $m_i(t)$  by Eq. (2)
16:   end for
17:   Get aggregated messages  $m'(t)$  by Eq. (3)
18:   Update region memory  $s(t)$  by Eq. (4)
19:   Generate region embedding  $Z'$  by Eq. (5)
20:   return Updated region memory  $s(t)$ , Region embedding  $Z'$ 
21: end procedure
22: procedure TIME-AWARE MOBILITY RECONSTRUCTION( $Z'_\theta, \mathbb{E}(t^-, t)$ )
23:   Get the set of negative region pairs  $\mathbb{E}^0(t^-, t)$ 
24:   for each positive pair  $(v_i, v_j, t_m) \in \mathbb{E}(t^-, t)$  do
25:     Get the predicted number of trips  $f^+(v_i, v_j, t_m)$  by (6)
26:   end for
27:   Get loss for negative pairs  $\mathcal{L}_{mob}^+$  by Eq. (7)
28:   for each negative pair  $(v_p, v_q) \in \mathbb{E}^0(t^-, t)$  do
29:     Get the predicted number of trips  $f^-(v_p, v_q)$  by (8)
30:   end for
31:   Get loss for negative pairs  $\mathcal{L}_{mob}^-$  by Eq. (9)
32:    $\mathcal{L}_{mob} = \mathcal{L}_{mob}^+ + \mathcal{L}_{mob}^-$ 
33:   return Mobility reconstruction loss  $\mathcal{L}_{mob}$ 
34: end procedure
35: procedure LONG-SCALE SAMPLING( $\mathbb{E}(t^-, t)$ )
36:   Get data of a longer scale  $\hat{\mathbb{E}}(t^-, t)$  by Eq. (11)
37:   Randomly sample edge  $\mathbb{E}_\xi(t^-, t)$  by Eq. (12)
38:   return  $\mathbb{E}_\xi(t^-, t)$ 
39: end procedure

```

- **RQ2:** How do different components (evolving embedding module, time-aware mobility reconstruction, cross-scale embedding reconstruction) affect the results?
- **RQ3:** What are the advantages of the evolving region representation? What can be reflected by the dynamic changing process of region representation?
- **RQ4:** Can the learned region embeddings representation detect movement patterns in human mobility data?

5.1. Data analysis

Datasets Real-world datasets are collected from two cities: **New York City**, which contain taxi orders and crime records in Manhattan from May to July in 2012,¹ **Chicago**, which contain taxi orders and crime records respectively in Chicago from May to July in 2016.² Detailed introductions of these datasets are shown in Table 1.

To better understand the downstream tasks, we performed data statistics for travel time data and crime data including the number of items, mean, standard deviation (SD), and zero ratio (the ratio of data whose crime count is 0).

¹ <https://opendata.cityofnewyork.us/>.

² <https://data.cityofchicago.org/>.

Table 1
Details of datasets.

City	New York City	Chicago
#Nodes	180	77
#Interactions	36,381,591	4,670,739
#Train days	42	42
#Validation days	7	7
#Test days	14	14

Table 2
Data statistics.

Data	Travel time	Crime
New York City	Number	36,381,591
	Mean	647.8171
	SD	415.6426
	Zero Ratio	27.47%
Chicago	Number	4,670,739
	Mean	856.4423
	SD	1064.0295
	Zero Ratio	0.35%

Table 2 shows that the standard deviation of travel time data is greater in Chicago than in New York City, indicating that Chicago travel time data are less stable, increasing the difficulty of prediction. Whereas in the crime data, New York City has significantly less data than Chicago, and 27.47% of the data is zero, suggesting that New York City's data quality is lower than Chicago's data quality. The sparseness of data makes it difficult to use region embeddings for crime prediction in New York City. The variance of experiment results between the two cities also validates the above analysis.

5.2. Baselines

Three types of baselines are compared. The first type is baselines for general urban region embedding tasks, such as REGION2VEC [44], MVURE [9], MGFN [42], HREP [11] and ROMER [45]. These methods mainly fuse multi-view mobility graphs to learn static general region embeddings. Baselines of the second type are task-specific baselines that deal with specific task scenarios in an end-to-end way. ST-SHN [46] is a crime prediction approach that constructs hypergraph structures to encapsulate broad spatial correlations. CTTE [47] is a travel time estimation method aggregating segment embeddings along the path. We also select some spatiotemporal prediction methods as baselines like GraphWaveNet [12] and MTGNN [13].

Baselines for general urban region embedding tasks. These methods mainly fuse multi-view mobility graphs to learn static general region embeddings, which subsequently are used in various downstream tasks.

- **REGION2VEC** [44] employs a multi-graph fusion module with an encoder-decoder framework to analyze relationships between regions, enhancing the learning of comprehensive region embeddings.
- **MVURE** [9] leverages data from within and between regions to construct graphs, and then applies multi-view integration to learn region embeddings.
- **MGFN** [42] builds human mobility patterns including direction and scale of flow through time from taxi trips to facilitate the learning of region embeddings.
- **HREP** [11] incorporates prompt learning to enhance the effectiveness of region embedding in downstream tasks.
- **ROMER** [45] captures multi-view dependencies among areas, avoiding strict neighborhood limitations, and then employs a dual-stage fusion module incorporating attention and gating techniques to generate region embeddings.

- **GraphST** [39] employs a cross-view contrastive learning paradigm to model interdependencies across view-specific region representations, preserving underlying relation heterogeneity for improved region embeddings.

Baselines for specific tasks. These methods are proposed to deal with specific task scenarios in an end-to-end way such as crime prediction, and travel time estimation.

- **ST-SHN** [46] is a crime prediction approach that constructs hypergraph structures to encapsulate the broad spatial correlations.
- **CTTE** [47] as a travel time estimation method aggregates segment embeddings along the path.

Baselines for spatio-temporal prediction. This type of approach predicts future feature values by modeling spatial and temporal relationships through historical sequential data. These models utilize a sequence of snapshots based on discrete time windows. Generally, these spatial-temporal prediction methods are usually used for predicting traffic flow data, we also use this type of approach for crime prediction given that crime prediction is also a time series prediction task.

- **GraphWaveNet** [12] integrates a flexible adjacency matrix within graph convolution alongside dilated causal convolution to model spatial and temporal relationships.
- **MTGNN** [13] is a spatial-temporal framework that leverages external features to create a one-way adaptive graph. It builds inter-variable relationships through the implementation of a graph-learning module.

5.3. Baseline settings

For general urban region embedding baselines like MVURE [9], the original paper constructs a graph using one year of data to generate region embeddings. In our implementation, we adapted the temporal granularity of the data based on the specific task requirements. For crime prediction, we built the graph using data segmented by prediction time steps, ensuring that region embeddings were learned accordingly during training. The downstream testing process remained consistent with the original paper, where the region embeddings from the last timestamp were concatenated with date information and fed into an MLP to generate the final prediction. For the travel time estimation task, the testing process involved concatenating the origin and destination region embeddings along with travel distance before feeding them into an MLP to estimate travel time. For traffic flow prediction, the region embeddings, along with the flow information from the history 7 timesteps, were fed into an MLP network to produce the final prediction.

For spatial-temporal methods like GraphWaveNet [12], these models do not generate explicit region representations, making them unsuitable for the travel time estimation task. Instead, they follow the standard temporal forecasting setup, where history 7 timesteps of historical data are used to predict the current value, which aligns with the typical baseline configurations for traffic flow and crime prediction tasks.

5.4. Experimental settings

The memory, message, and embedding dimensions are 144 for both datasets. The temporal scaling factor α is chosen from [1,2,3], the β is chosen from 0.1 to 0.5 and the decay rate γ is chosen from [0.9,0.99,0.999]. The proposed model is trained using the Adam optimizer set with a starting learning rate of 0.0001, along with an early stopping approach that has the patience of 20. The learning rate for the various deep learning techniques is selected from the values [0.01, 0.001, 0.0001, 0.00001] based on optimal performance on the validation set. The parameters that yield the best results during validation are then used to assess the performance on the test set. The comparison metrics chosen include Mean Average Error (MAE), Root Mean Square Error (RMSE), and Pearson Correlation Coefficient (PCC).

5.5. Dynamic crime prediction (RQ1)

Dynamic crime prediction aims to predict the total number of regional crimes in a day. Region embeddings of the last timestamp are concatenated with the date info as input of MLP to get the final prediction.

The findings are presented in Table 3. It is found that EvolveURE performs the best compared to baselines of three types.

Besides, we plot the predictions of EvolveURE and HREP for the real crime count on a given day in Fig. 5. It can be observed that our method predicts the count of crimes closer to the ground truth than HREP's. The result shows that the embeddings of the regions obtained from our model learning better reflect the function and properties of the region.

5.6. Travel time estimation (RQ1)

For travel time estimation, we concatenate origin region embedding with destination region embedding plus travel distance and then feed them into MLP to estimate the travel time. Table 4 shows an obvious superiority on the travel time estimation task, which indicates that acquiring fine-grained information on the edge and evolving region embeddings can enhance performance on challenging downstream tasks that involve intricate dynamics.

5.7. Traffic flow prediction (RQ1)

The goal of traffic flow prediction is to forecast the traffic flow for each region over a half-hour period. For traffic flow prediction, the region embeddings connected with the flow information of the last 7 time steps are fed into the MLP network to get the prediction. As shown in Table 5, spatial-temporal models perform better than other baselines on flow prediction.

5.8. Ablation study (RQ2)

In order to examine the impact of each component in EvolveURE, the proposed model is contrasted against three different variants:

- **w/o Dynamic** removes the evolving region embedding module and uses static embeddings as the final generated region embeddings.
- **w/o Time** removes the time-aware mobility graph reconstruction module.
- **w/o Cross** removes the cross-scale embedding reconstruction module.

Ablation studies are conducted on crime prediction and travel time estimation. Crime prediction and travel time estimation are two representation downstream tasks in inter-region downstream tasks and intra-region downstream tasks separately.

From Fig. 6, it can be observed that the evolving region embedding module plays the most critical role in EvolveURE among all modules. Removing the evolving region embeddings causes a drastic drop in performance. The experimental result of w/o Time and w/o Cross varies on two tasks. It indicates that the task of travel time estimation requires more detailed temporal information while the crime prediction benefits more from removing data noises. The performance drop is more obvious in travel time estimation when removing one component. Overall, all components are indispensable, while the evolving region embedding module plays the most crucial role in keeping the performance.

We also further conducted an ablation experiment on the task of flow prediction. Results are drawn in Fig. 7.

Table 3
Experiment results on dynamic crime prediction.

	Task	Dynamic crime prediction		
		MAE	RMSE	PCC
New York City	REGION2VEC	1.4243 ± 0.0030	2.0604 ± 0.0051	0.5614 ± 0.0020
	MVURE	1.1175 ± 0.0012	1.6509 ± 0.0065	0.7444 ± 0.0012
	MGFN	1.0993 ± 0.0010	1.6318 ± 0.0080	0.7509 ± 0.0004
	HREP	1.2471 ± 0.0019	1.7752 ± 0.0056	0.7007 ± 0.0002
	ROMER	1.2826 ± 0.0016	1.8454 ± 0.0092	0.6712 ± 0.0056
	GraphST	1.0956 ± 0.0016	1.6108 ± 0.0052	0.7512 ± 0.0010
	ST-SHN	1.5965 ± 0.0092	2.0681 ± 0.0137	0.6648 ± 0.0044
	GraphWaveNet	1.7167 ± 0.0118	2.4225 ± 0.0221	0.7386 ± 0.0033
	MTGNN	1.6502 ± 0.0012	2.4359 ± 0.0329	0.7495 ± 0.0028
Chicago	EvolveURE	1.0657 ± 0.0042	1.6145 ± 0.0066	0.7552 ± 0.0001
	REGION2VEC	4.2989 ± 0.0532	5.7728 ± 0.0675	0.7982 ± 0.0054
	MVURE	2.8322 ± 0.0418	3.9507 ± 0.0583	0.9128 ± 0.0042
	MGFN	3.0353 ± 0.0467	4.3095 ± 0.0641	0.9053 ± 0.0038
	HREP	3.8148 ± 0.0502	6.2346 ± 0.0729	0.7644 ± 0.0061
	ROMER	4.2398 ± 0.0489	4.8798 ± 0.0617	0.6875 ± 0.0047
	GraphST	2.8586 ± 0.0431	4.1759 ± 0.0604	0.9082 ± 0.0044
	ST-SHN	2.7355 ± 0.0395	3.7625 ± 0.0556	0.9208 ± 0.0039
	GraphWaveNet	3.0267 ± 0.0458	4.3981 ± 0.0623	0.8958 ± 0.0046
Chicago	MTGNN	2.8145 ± 0.0407	3.9511 ± 0.0581	0.9153 ± 0.0041
	EvolveURE	2.6552 ± 0.0193	3.7010 ± 0.0340	0.9225 ± 0.0015

Table 4
Experiment results on travel time estimation.

	Task	Travel time estimation		
		MAE	RMSE	PCC
New York City	REGION2VEC	172.4262 ± 0.4249	235.1823 ± 0.7796	0.7790 ± 0.0014
	MVURE	170.1046 ± 0.3116	232.8493 ± 0.5443	0.7838 ± 0.0027
	MGFN	169.4540 ± 0.2882	232.0928 ± 0.3149	0.7857 ± 0.0015
	HREP	172.6659 ± 0.7113	236.6504 ± 0.2987	0.7755 ± 0.0008
	ROMER	168.8000 ± 0.3222	233.1593 ± 0.4122	0.7828 ± 0.0023
	GraphST	168.4600 ± 0.2112	232.4263 ± 0.2973	0.7911 ± 0.0048
	CTTE	156.3153 ± 0.1066	221.8708 ± 0.2835	0.8060 ± 0.0012
	EvolveURE	136.2585 ± 0.3778	192.4681 ± 0.5182	0.8583 ± 0.0033
Chicago	REGION2VEC	393.3553 ± 0.0512	467.8522 ± 0.3456	0.6170 ± 0.0006
	MVURE	393.2955 ± 0.0308	466.5886 ± 0.2341	0.6195 ± 0.0002
	MGFN	393.6484 ± 0.0007	466.9496 ± 0.1324	0.6190 ± 0.0003
	HREP	396.1964 ± 0.1223	471.3260 ± 1.1312	0.6098 ± 0.0001
	ROMER	396.5959 ± 0.0132	469.7240 ± 0.3415	0.6127 ± 0.0014
	GraphST	393.2889 ± 0.3021	466.2114 ± 0.3527	0.6192 ± 0.0004
	CTTE	409.2653 ± 0.2218	475.7073 ± 0.2453	0.6006 ± 0.0013
	EvolveURE	385.9561 ± 0.1219	453.9497 ± 0.6533	0.6458 ± 0.0007

Table 5
Experiment results on traffic flow prediction.

	Task	Traffic flow prediction		
		MAE	RMSE	PCC
New York City	REGION2VEC	7.8342 ± 0.0320	14.3634 ± 0.0693	0.9772 ± 0.0003
	MVURE	7.5827 ± 0.0231	13.8867 ± 0.0321	0.9787 ± 0.0002
	MGFN	7.4509 ± 0.0039	13.6489 ± 0.0123	0.9795 ± 0.0002
	HREP	7.9402 ± 0.0227	14.4511 ± 0.0402	0.9770 ± 0.0004
	ROMER	7.6714 ± 0.0114	14.0486 ± 0.0397	0.9782 ± 0.0003
	GraphST	7.3989 ± 0.0105	13.2887 ± 0.0289	0.9811 ± 0.0002
	GraphWaveNet	6.5354 ± 0.0298	12.2418 ± 0.0324	0.9842 ± 0.0005
	MTGNN	6.2882 ± 0.0171	11.7505 ± 0.0295	0.9850 ± 0.0004
	EvolveURE	6.1744 ± 0.0083	11.2383 ± 0.0193	0.9861 ± 0.0002
Chicago	REGION2VEC	2.5951 ± 0.0078	9.0440 ± 0.0007	0.9942 ± 0.0001
	MVURE	2.6144 ± 0.0029	9.0451 ± 0.0031	0.9942 ± 0.0002
	MGFN	2.5394 ± 0.0032	8.9296 ± 0.0042	0.9943 ± 0.0001
	HREP	2.6145 ± 0.0021	9.0256 ± 0.0088	0.9942 ± 0.0001
	ROMER	2.5775 ± 0.0034	9.1492 ± 0.0031	0.9940 ± 0.0001
	GraphST	2.5127 ± 0.0045	8.7796 ± 0.0018	0.9951 ± 0.0003
	GraphWaveNet	2.5040 ± 0.0097	9.1679 ± 0.0039	0.9947 ± 0.0001
	MTGNN	2.4028 ± 0.0013	8.5673 ± 0.0043	0.9948 ± 0.0002
	EvolveURE	2.2573 ± 0.0056	7.7965 ± 0.0032	0.9957 ± 0.0005

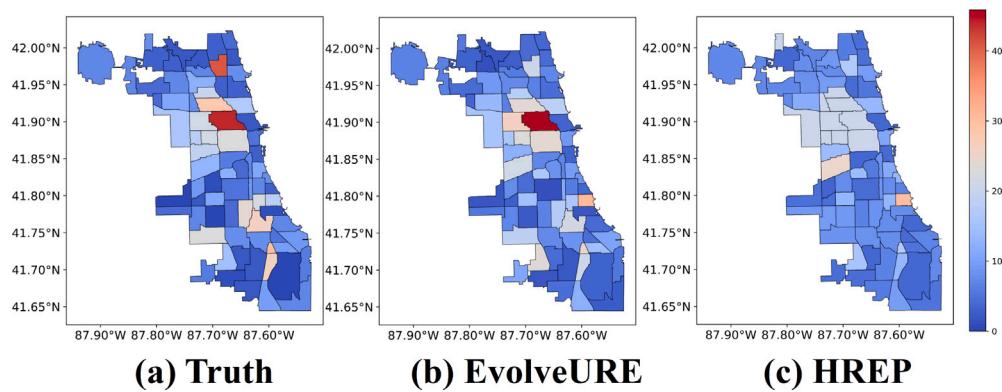


Fig. 5. Region distribution of crime count on Chicago. The redder the color, the higher the number of crimes.

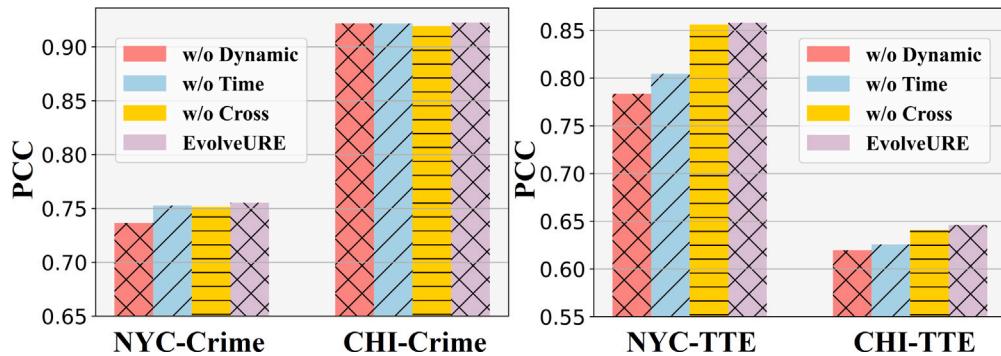


Fig. 6. Ablation studies

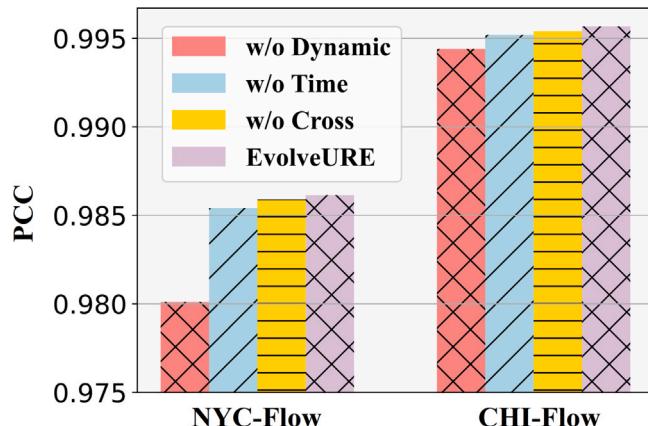


Fig. 7. Ablation study in traffic flow prediction task.

5.9. The dynamics of region embeddings (RQ3)

To study the evolutionary trend of the embedding, embeddings of a region in one day are processed by PCA and illustrated in Fig. 8(a), with the starting point in green and the endpoint in orange. Fig. 8(b) shows the curve of traffic flow during the same day. First, the evolutionary trend of embedding in a day tends to be close to the ring, which indicates that the region is characterized by similar attributes at the same moment on different days. Second, the evolution of the embeddings keeps consistency with the traffic flow curve: when the embedding moves to the upper right in the representation space, the traffic flow decreases (00:00 to 06:00). When the traffic flow rises at noon, the embedding also moves to the lower left (8:00 to 12:00). In the evening, when the traffic flow declines, the embedding moves to

the upper right again, gradually returning to its starting position (after 18:00).

The above example shows that region embeddings obtained by EvolveURE imply evolutionary properties of regions throughout the day and can support fine-grained and continuous dynamic urban downstream tasks, which was previously not possible with static region embeddings.

5.10. The process of embedding evolving (RQ3)

To intuitively illustrate the evolving process of region embedding, we compress region embeddings at different times during one day to 2-dimensional vectors via t-SNE and visualize six selected regions and their positions in Fig. 9.

It can be observed in Fig. 9(a) that regions marked with orange (138, 140, 162) are adjacent to each other while regions marked with red (24, 89, 155) are geographically proximate. We visualize their embeddings from different hours in Fig. 9(b). Geographically neighboring regions are also adjacent in the representation space, which holds for both red and orange regions. Besides, the function of a region can be reflected in the difference between inflows and outflows, e.g. residential areas usually have more outflows in the morning and more inflows in the evening, while commercial areas have the opposite traffic flow characteristics.

We count the difference between inflows and outflows on the same day for the red and orange regions in [Table 6](#). Between 10:00 and 20:00, the red and orange regions reflect the similar traffic flow patterns of the residential areas which make them gather in representation space. However, at midnight (0:00 and 24:00), the orange regions have more inflows than outflows, reflecting residential properties, while the red regions have more outflows than inflows, consistent with commercial properties. Function differences at this time lead to their dispersion in the representation space. The above cases validate that embedding obtained by EvolveURE can evolve to reflect the different functions of the region at different times.

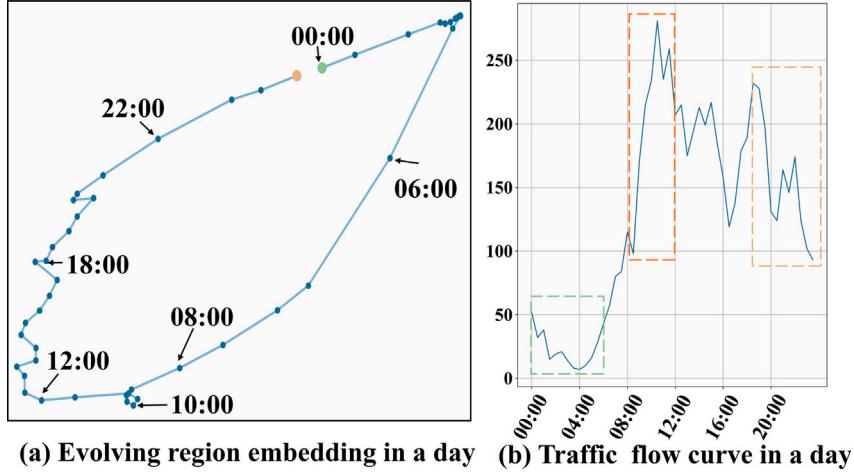


Fig. 8. The dynamics of the learned region embeddings.

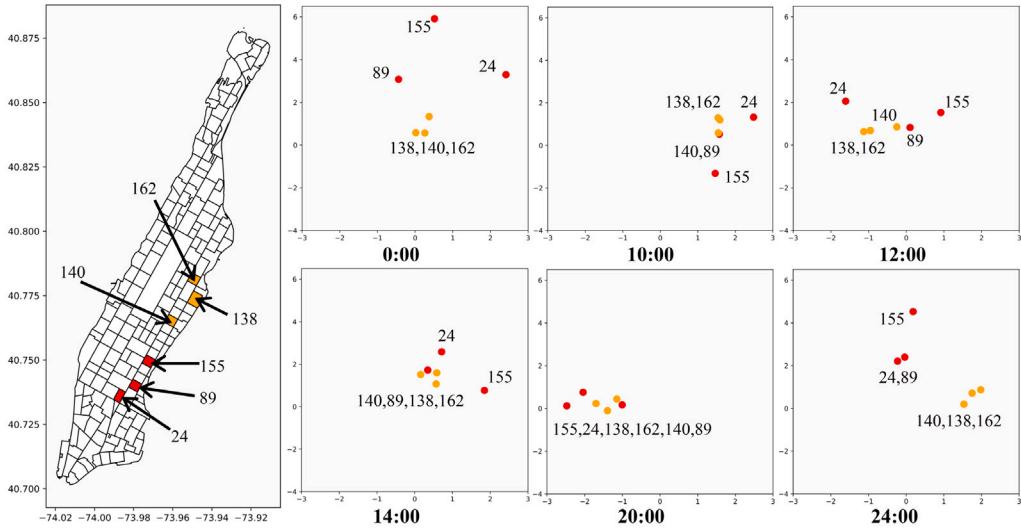


Fig. 9. Illustration of embedding evolving process.

Table 6
The count of inflows minus outflows.

Region	Hours					
	0:00	10:00	12:00	14:00	20:00	24:00
Red	-31	-88	-114	-79	61	-15
Orange	157	-113	-126	-28	151	63

5.11. Finding different mobility patterns (RQ4)

To gain a deeper insight into the correlation between the learned region embeddings, a case study is conducted in New York City. The region embeddings obtained in one day are transformed into 2-dimensional vectors using t-SNE and then grouped into several clusters through K-Means.

Fig. 10 illustrates that regions with analogous traffic flow curves tend to be nearer in the representation space. For instance, R1 and R2 exhibit a similar traffic pattern (there is a period of obvious rush hours each day and this peak drops to a very low level on weekends) and they are situated within the same clusters, while O1 and O2 are alike (traffic flow peaks are roughly the same during a week).

The consistency between the learned region embedding and the traffic flow curve of the regions indicates that EvolveURE can automatically extract useful information from the original data and compress it into significant embeddings.

5.12. Parameter sensitivity analysis

We examine how variations in the memory/message dimension, learning rate, temporal scaling factor α , and quantity scaling factor β influence the experimental outcomes. Experiment results have been displayed in Fig. 11. PCC is employed on the travel time estimation task to measure performance. Among the chosen list of [64, 128, 144, 256, 512], the best dimension of memory/message is 144. The learning rate significantly influences model performance in comparison to other parameters. The model performs optimally with a learning rate of 0.0001. Changes in the temporal scaling factor and the quantity scaling factor have a small effect on the experiment results. We select the parameter configurations that yield the best results within the range as the final settings for the experiment.

6. Conclusion

This work emphasizes the development of evolving representations for various regions to enhance dynamic urban sensing tasks.

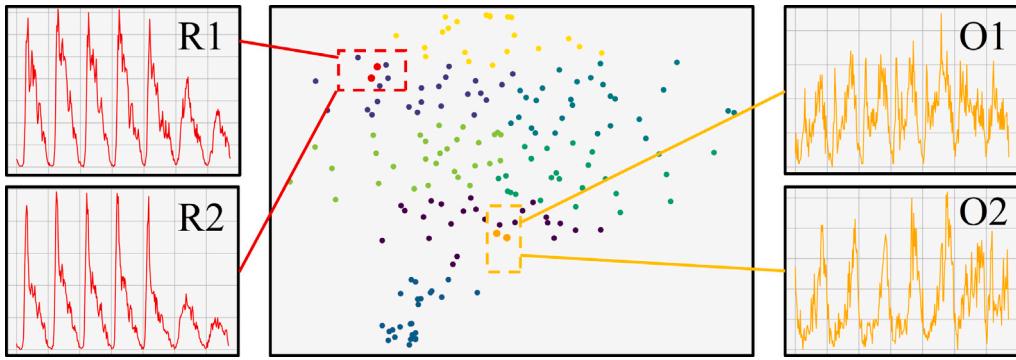


Fig. 10. Region embeddings clustering results and traffic flow curves. In the middle part of the figure is the result of the clustering of the 180 region embeddings (each point represents a vector), and on both sides of the figure are the traffic flow curves for the corresponding selected regions for one week.

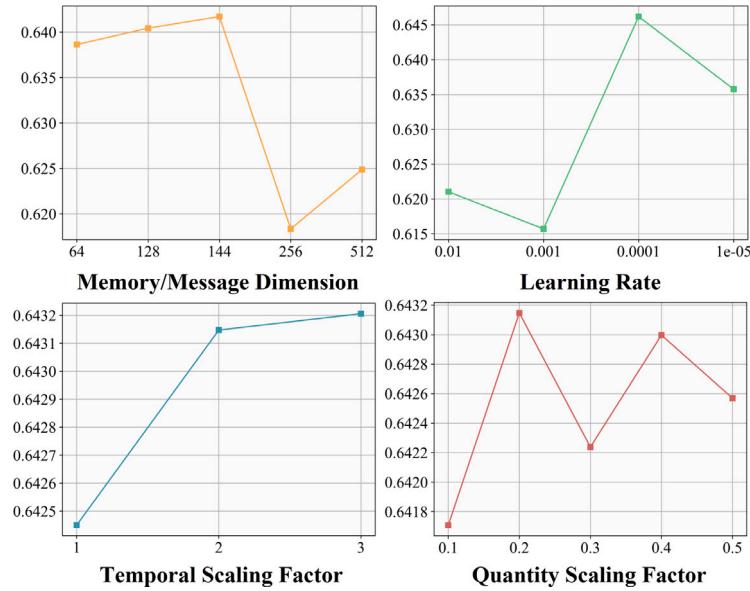


Fig. 11. Parameter sensitivity analysis.

We designed: (1) an evolving region embedding encoder to encode the region-region interactions chronologically and continuously update the memories that store the dynamic states of regions; (2) a time-aware mobility reconstruction decoder to reconstruct dynamic mobility graphs with the generated evolving embeddings; (3) a cross-scale embedding module designed to enhance stability in the embedding process and minimize data noise. Extensive experiments with multiple downstream tasks verified the effectiveness and necessity of dynamic region embedding.

7. Practical applications in urban planning and computing

The proposed urban region embedding framework has significant potential for widespread application in urban planning and computing. In urban planning, the evolving region embeddings can be utilized to optimize land use by identifying functional similarities and differences between regions. For example, planners can leverage the embeddings to allocate resources more efficiently by analyzing regions with similar functional characteristics (e.g., commercial, residential, or industrial zones). Additionally, the embeddings can support transportation network design by analyzing mobility patterns and interaction scores between regions, enabling the identification of areas that require improved connectivity through new roads or public transit routes. In urban computing, the framework can be applied to region classification and recommendation tasks, such as business site selection or real estate

recommendation. By analyzing the embedding vectors, businesses can identify regions with similar consumer behavior patterns for targeted marketing strategies. Furthermore, the embeddings can be used for anomaly detection by monitoring changes in regional functionality over time, such as detecting sudden shifts from residential to commercial areas, which can inform timely urban management decisions.

8. Limitations and future work

However, this paper also has some limitations. This paper primarily focuses on addressing dynamic urban sensing tasks. Accordingly, the designed methods and updating mechanisms are specifically tailored to solve challenges associated with dynamic tasks, which forms the core contributions of this work. Currently, we use static embedding vectors to learn urban attributes and update them through gradient descent. In the future, we plan to introduce more advanced designs in this area and conduct tests across a broader range of urban sensing task scenarios. Additionally, we aim to explore the integration of multi-modal data (e.g., satellite imagery, social media data) to further enhance the model's applicability and collaborate with urban planning authorities to implement the framework in real-world smart city initiatives.

CRediT authorship contribution statement

Yi Xu: Methodology. **Zerong Deng:** Validation. **Tongyu Zhu:** Formal analysis. **Liangzhe Han:** Writing – review & editing. **Leilei Sun:** Supervision. **Zhuo Chen:** Funding acquisition. **Hao Sheng:** Software.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62272023, No. U24B20171, No. 62394332) and Research Start-up Funds of Hangzhou International Innovation Institute of Beihang University (No. 2024KQ053, No. 2024KQ068).

Data availability

Data will be made available on request.

References

- [1] J. Wang, X. Kong, F. Xia, L. Sun, Urban human mobility: Data-driven modeling and prediction, ACM SIGKDD Explor. Newsl. 21 (1) (2019) 1–19.
- [2] S. Ji, Y. Zheng, T. Li, Urban sensing based on human mobility, in: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 2016, pp. 1040–1051.
- [3] C. Huang, J. Zhang, Y. Zheng, N.V. Chawla, DeepCrime: Attentive hierarchical recurrent networks for crime prediction, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1423–1432.
- [4] L. Han, B. Du, L. Sun, Y. Fu, Y. Lv, H. Xiong, Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 547–555.
- [5] D. Wang, J. Zhang, W. Cao, J. Li, Y. Zheng, When will you arrive? estimating travel time based on deep neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
- [6] C. Huang, C. Zhang, P. Dai, L. Bo, Cross-interaction hierarchical attention networks for urban anomaly prediction, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4359–4365.
- [7] H. Wang, Z. Li, Region representation learning via mobility flow, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 237–246.
- [8] Z. Yao, Y. Fu, B. Liu, W. Hu, H. Xiong, Representing urban functions through zone embedding with human mobility patterns, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, 2018.
- [9] M. Zhang, T. Li, Y. Li, P. Hui, Multi-view joint graph representation learning for urban region embedding, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4431–4437.
- [10] Y. Zhang, Y. Xu, L. Cui, Z. Yan, Multi-view graph contrastive learning for urban region representation, in: 2023 International Joint Conference on Neural Networks, IJCNN, IEEE, 2023, pp. 1–8.
- [11] S. Zhou, D. He, L. Chen, S. Shang, P. Han, Heterogeneous region embedding with prompt learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 4981–4989.
- [12] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph WaveNet for deep spatial-temporal graph modeling, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10–16, 2019, ijcai.org, 2019, pp. 1907–1913.
- [13] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 753–763.
- [14] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, 2017, arXiv preprint arXiv:1707.01926.
- [15] J. Chen, X. Wang, X. Xu, GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction, Appl. Intell. (2022) 1–16.
- [16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, 2014, arXiv preprint arXiv:1406.1078.
- [17] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Adv. Neural Inf. Process. Syst. 28 (2015).
- [18] X. Zhu, Y. Wu, L. Wang, H. Su, Z. Li, Continuous-time dynamic interaction network learning based on evolutionary expectation, IEEE Trans. Cogn. Dev. Syst. 16 (3) (2023) 840–849.
- [19] Z.-P. Li, H.-L. Su, X.-B. Zhu, V. Gribova, V.F. Filaretov, D.-S. Huang, SSPool: A simple siamese framework for graph infomax pooling, IEEE Trans. Netw. Sci. Eng. (2023).
- [20] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 701–710.
- [21] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al., Graph attention networks, Stat 1050 (20) (2017) 10–48550.
- [22] A. Grover, J. Leskovec, Node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.
- [23] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.
- [24] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Adv. Neural Inf. Process. Syst. 30 (2017).
- [25] Y. Xu, L. Han, T. Zhu, L. Sun, B. Du, W. Lv, Generic dynamic graph convolutional network for traffic flow forecasting, Inf. Fusion 100 (2023) 101946.
- [26] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 922–929.
- [27] C. Zheng, X. Fan, C. Wang, J. Qi, Gman: A graph multi-attention network for traffic prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 1234–1241.
- [28] S. Kumar, X. Zhang, J. Leskovec, Predicting dynamic embedding trajectory in temporal interaction networks, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1269–1278.
- [29] R. Trivedi, M. Farajtabar, P. Biswal, H. Zha, Dyrep: Learning representations over dynamic graphs, in: International Conference on Learning Representations, 2019.
- [30] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, M. Bronstein, Temporal graph networks for deep learning on dynamic graphs, 2020, arXiv preprint arXiv:2006.10637.
- [31] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, K. Acham, Inductive representation learning on temporal graphs, 2020, arXiv preprint arXiv:2002.07962.
- [32] W. Cong, S. Zhang, J. Kang, B. Yuan, H. Wu, X. Zhou, H. Tong, M. Mahdavi, Do we really need complicated model architectures for temporal networks?, 2023, arXiv preprint arXiv:2302.11636.
- [33] M.C. Gonzalez, C.A. Hidalgo, A.-L. Barabasi, Understanding individual human mobility patterns, Nature 453 (7196) (2008) 779–782.
- [34] H. Barbosa, M. Barthelemy, G. Ghoshal, C.R. James, M. Lenormand, T. Louail, R. Menezes, J.J. Ramasco, F. Simini, M. Tomassini, Human mobility: Models and applications, Phys. Rep. 734 (2018) 1–74.
- [35] W. Dang, H. Wang, S. Pan, P. Zhang, C. Zhou, X. Chen, J. Wang, Predicting human mobility via graph convolutional dual-attentive networks, in: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, 2022, pp. 192–200.
- [36] T. Huang, Z. Wang, H. Sheng, A.Y. Ng, R. Rajagopal, M3G: Learning urban neighborhood representation from multi-modal multi-graph, Proc. Deep. 2021 (2021) 2nd.
- [37] N. Kim, Y. Yoon, Effective urban region representation learning using heterogeneous urban graph attention network (HUGAT), 2022, arXiv preprint arXiv:2202.09021.
- [38] Q. Zhang, C. Huang, L. Xia, Z. Wang, Z. Li, S. Yiu, Automated spatio-temporal graph contrastive learning, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 295–305.
- [39] Q. Zhang, C. Huang, L. Xia, Z. Wang, S.M. Yiu, R. Han, Spatial-temporal graph learning with adversarial contrastive adaptation, in: International Conference on Machine Learning, PMLR, 2023, pp. 41151–41163.
- [40] Z. Li, W. Huang, K. Zhao, M. Yang, Y. Gong, M. Chen, Urban region embedding via multi-view contrastive prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 8724–8732.
- [41] Z.-P. Li, S.-G. Wang, Q.-H. Zhang, Y.-J. Pan, N.-A. Xiao, J.-Y. Guo, C.-A. Yuan, W.-J. Liu, D.-S. Huang, Graph pooling for graph-level representation learning: a survey, Artif. Intell. Rev. 58 (2) (2024) 45.
- [42] S. Wu, X. Yan, X. Fan, S. Pan, S. Zhu, C. Zheng, M. Cheng, C. Wang, Multi-graph fusion networks for urban region embedding, 2022, arXiv preprint arXiv:2201.09760.
- [43] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, 2015, arXiv preprint arXiv:1509.02971.

- [44] Y. Luo, F.-l. Chung, K. Chen, Urban region profiling via multi-graph representation learning, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 4294–4298.
- [45] W. Chan, Q. Ren, Region-wise attentive multi-view representation learning For Urban Region embedding, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 3763–3767.
- [46] L. Xia, C. Huang, Y. Xu, P. Dai, L. Bo, X. Zhang, T. Chen, Spatial-temporal sequential hypergraph network for crime prediction with dynamic multiplex relation learning, 2022, arXiv preprint [arXiv:2201.02435](https://arxiv.org/abs/2201.02435).

- [47] R. Gao, F. Sun, W. Xing, D. Tao, J. Fang, H. Chai, CTTE: customized travel time estimation via mobile crowdsensing, *IEEE Trans. Intell. Transp. Syst.* 23 (10) (2022) 19335–19347.